TÓM LƯỢC BÀI GIẢNG

(Vũ Quốc Hoàng)

CON TRỎ VÀ MẢNG, THAM CHIẾU (C++)

Chủ đề

- Con trỏ và mảng
- Tham chiếu (C++)

Tài liệu

[1] Paul Deiteil, Harvey Deiteil, *C how to program*, Pearson, 8th global edition, 2016.

[2] Tony Gaddis, *Starting out with C++ From Control Structures through Objects*, Pearson, 8th edition, 2015.

Đọc tài liệu

- Con trỏ và mảng: Mục 7.4, 7.8 và 7.9 [1]
- Tham chiếu (C++): Mục 6.13 [2]

Kiến thức

- Đoạn mã và hình minh họa 1 (lưu ý là các địa chỉ cụ thể 0x62d0, … chỉ có tính minh họa)

```
int a[] = {1, 2, 3, 4};
int *p = a;
*(p + 2) = 5;

printf("Gia tri phan tu thu 0: %d %d\n",
            a[0], *p);
printf("Dia chi phan tu thu 0: %x %x\n",
            &a[0], a);
printf("Gia tri phan tu thu 2: %d %d\n",
            a[2], *(a + 2));
printf("Dia chi phan tu thu 2: %x %x\n",
            &a[2], p + 2);
printf("What's this?: %d %d\n", p[0], p[2]);

char s[] = "hello";
char *r = s;
r += 2;
*r = 'h'; r[1] = 'i';
r[-1] = 'i'; r[2] = '\0';
printf("What's this?: %x %x %s\n", s, r, s);
```

a, p   p+2

0x62d0   0x62d4   0x62d8   0x62dc

… | 1 | 2 | ~~3~~ 5 | 4 | …

a[0]   a[1]   a[2]   a[3]

p   r

… | 'h' | 'i' ~~'e'~~ | 'h' ~~'l'~~ | 'i' ~~'l'~~ | 0 ~~'o'~~ | 0 | …

0x82d0   0x82d1   0x82d2   0x82d4

s, ~~r~~   r-1   r   r+2

- Mảng/chuỗi có mối quan hệ mật thiết với con trỏ trong C/C++: tên mảng/chuỗi là địa chỉ của phần tử đầu tiên của mảng/chuỗi và có thể được xem là <u>hằng con trỏ</u>.
- Con trỏ và mảng cũng gần gũi về mặt kí pháp. Khi a là tên mảng hay con trỏ cùng kiểu, `index` là một biểu thức nguyên thì: `a[index]` tương đương với `*(a + index)`, còn `&a[index]` tương đương với `(a + index)`.

- *Số học con trỏ* (pointer arithmetic) liên quan đến các phép toán trên con trỏ. Trong đó quan trọng nhất là phép cộng trừ số nguyên trên con trỏ và tăng/giảm con trỏ (xem là cộng/trừ 1). Khi con trỏ p được khai báo là trỏ đến vùng nhớ kiểu *T* (tức là khai báo `T *p`) và p đang chứa địa chỉ là *X* thì biểu thức `(p + i)` sẽ là địa chỉ *X* + *i*×sizeof(*T*) với `i` là một biểu thức nguyên. Điều này rất dễ hình dung khi nhớ rằng `(p + i)` tương đương với `&p[i]` mà về trực giác có nghĩa là: nếu p là địa chỉ phần tử đầu tiên của một mảng (kiểu *T*) thì `(p + i)` là địa chỉ của phần tử thứ `i` của mảng đó.
- Đoạn mã và hình minh họa 2 (lưu ý là các địa chỉ cụ thể 0x62d0, … chỉ có tính minh họa)

```
int a;
int &b = a;
int *p = &a;

a = 10; b = 20; *p = 30;

printf("Gia tri cua a: %d %d %d\n",
            a, b, *p);
printf("Dia chi cua a: %x %x %x\n",
            &a, &b, p);
```


- C++ có một dạng biến đặc biệt nữa là *biến tham chiếu* (reference variable), được khai báo với dấu & trước tên biến. Biến tham chiếu giúp đặt tên khác (alias) cho vùng nhớ của một biến đã khai báo. Sau khi được gán, biến tham chiếu được dùng như biến thông thường.
- Đoạn mã và hình minh họa 3 (lưu ý là các địa chỉ cụ thể 0x62d0, … chỉ có tính minh họa)

```
void func1(int v1)
{
    v1++;
}
void func2(int &v2)
{
    v2++;
}
void func3(int *v3)
{
    (*v3)++;
}

int main()
{
    int a, b, c;
    a = b = c = 1;

    printf("Gia tri truoc: %d %d %d\n",
            a, b, c);
    func1(a); func2(b); func3(&c);
    printf("Gia tri sau: %d %d %d\n",
            a, b, c);
}
```

- *Tham số* (parameter) của hàm có thể được xem là biến đặc biệt của hàm: nó được gán giá trị từ *đối số* (argument) trong lời gọi hàm và sau đó được dùng như *biến cục bộ* (local variable) khi hàm thực thi. Do đó, trong C/C++ ta có 3 loại tham số là: *"tham trị"* (các tham số bình thường), *tham trỏ* (pointer parameter, tham số kiểu con trỏ) và *tham chiếu* (reference parameter, tham số dạng tham chiếu). Để có thể thay đổi được dữ liệu chứa trong đối số (khi đối số là *l-value*) thì hàm cần dùng tham trỏ hoặc tham chiếu. Khi đó, việc truyền/nhận đối số của hàm được gọi là *truyền tham chiếu* (pass-by-reference). Với tham trị thì việc truyền/nhận đối số được gọi là *truyền giá trị* (pass-by-value).

Kĩ năng

- Thành thạo các kí pháp tương đương giữa mảng và con trỏ
- Thành thạo số học con trỏ
- Biết cách khai báo tham trỏ, tham chiếu và vận dụng được tham trỏ, tham chiếu khi muốn thay đổi giá trị của đối số trong hàm

Lưu ý

- Con trỏ là đặc trưng, kĩ thuật mạnh mẽ, quan trọng và "hơi khó" kiểm soát của C/C++ nên sinh viên cần rèn luyện nhiều để quen thuộc và thành thạo

Bài tập

Các bài tập (làm được) trong Chapter 7 [1]